800.0015
A01455

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Barry et al.
Serial No.:	10/815,294
Filed:	April 1, 2004
For:	METHODS AND APPARATUS FOR ADDRESS TRANSLATION FUNCTIONS
Group:	2188
Examiner:	Doan, Duc T.

Durham, North Carolina
March 27, 2006

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATION OF FACSIMILE TRANSMISSION

Sirs:

I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office, Fax. No. 571-273-8300 on the date set forth below

1. Transmittal of Appellants' Brief;
2. PTO-2038 Credit Card Payment Form (1 page); and
3. Appeal Brief (32 pages).

Marianna Tortorelli
Printed name of person signing

Marianna Tortorelli
Signature

Date: March 27, 2006

800.0015
A01455

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Barry et al.
For	:	Methods and Apparatus for Address Translation Functions
Serial No.	:	10/815,294
Filed	:	04/01/2004
Group	:	2188
Examiner	:	Doan, Duc T.

Durham, North Carolina
March 27, 2006

MAIL STOP APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPELLANT'S BRIEF

Dear Sirs:

1. Transmitted herewith is the APPEAL BRIEF in this application with respect to the Notice of Appeal filed on January 29, 2007.

2. The Applicant is other than a small entity.

3. Pursuant to 37 CFR 1.17(f) the fee for filing the Appeal Brief is $500.00.

   [x] The Commissioner is hereby authorized to charge the fee of $500 to our credit card. A credit card form is attached.

   [x] The Commissioner is hereby authorized to charge any additional fees which may be required or credit any overpayment to Law Offices of Peter H. Priest Deposit Account No. 50-1058.

Respectfully submitted,

Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713
(919) 806-1600

800.0015
A01455

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Barry et al.
For	:	Methods and Apparatus for Address Translation Functions
Serial No.	:	10/815,294
Filed	:	04/01/2004
Group	:	2188
Examiner	:	Doan, Duc T.

Durham, North Carolina
March 27, 2007

MAIL STOP APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF

Sir:

1. The Real Party In Interest

The real party in interest is the assignee, Altera Corporation.

2. Related Appeals and Interferences

None.

3. Status of the Claims

This is an appeal from the October 30, 2006 final rejection of claims 1-19, all of the pending claims. Claim 1 was rejected under 35 U.S.C. § 102(b) based on Saulsbury et al. U.S. Patent Publication No. 2002/0032710 (Saulsbury). Claim 2 was rejected under 35 U.S.C. § 103(a) as unpatentable over Saulsbury as applied to claim 1. Claims 3-19 were rejected under 35 U.S.C. § 103(a) as unpatentable over Saulsbury in view of Nair et al. U.S. Patent No. 6,944,747 (Nair). Pending claims 1-19 are the subject of this appeal.

4. Status of Amendments

The claims stand as last amended on August 21, 2006. No Amendment After-Final has been filed.

5. Summary of Claimed Subject Matter

The present invention relates to address translation apparatuses and methods for translating an address of a data element to a different address according to a translation pattern or a function. Figs. 1-10 and the text at page 5 et seq. provide a detailed description of the invention.

Claim 1

More particularly, claim 1 addresses a "processor address translation apparatus for translating an instruction operand address to a different operand address." As discussed at page 5, lines 16-18, a processor 100 shown in Fig. 1 may be adapted for use in conjunction with various embodiments of the present invention. Fig. 2D shows a processor subsystem 230 having an Rx read port translator 232 for translating instruction operand addresses to different addresses for reading addressable data from a register file 238 as described in detail at page 9, line 15 – page 10, line 15.

The apparatus of claim 1 comprises "a memory with an address input for selecting a data element from a plurality of data elements." The processor subsystem 230 of Fig. 2D shows a memory as the register file 238 having an address port 240 input that is supplied by the output of a port address register 239. A 1 to 8 selector 241 is used to decode the binary input provided on the address port 240 into one of eight selection signals 242 for selecting a data element from a plurality of data elements R0-R7 as described at page 10, lines 1-3.

The apparatus of claim 1 also comprises "an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution." Instruction register (IR) 108 of Fig. 1 receives instructions from an instruction bus 106 that is coupled to a short instruction word memory 104. As an example, an arithmetic instruction is received in the IR 108 having three register file address fields, Rt 142, Rx 144, and Ry 146. The arithmetic instruction received in the IR 108 further supplies opcode and control bits 120 over an opA bus 154 to a decode and control unit 118 as described at page 6, lines 3-11 and at page 7, lines 11-22.

The apparatus of claim 1 further comprises "an address translation unit for accessing the memory in a translation pattern, having the operand address as input and, in response to the instruction received in the instruction register, translating the operand address to form the different operand address in accordance with the translation pattern, the different operand address accessing a data element from the memory through the address input." In greater detail, the processor subsystem 230 of Fig. 2D shows an instruction register (IR) 231 that supplies an operand address bus 236 to an address translator 232. The address translator 232 also receives a control input 252 from a decode and control unit, such as decode and control unit 118 of Fig. 1, and a load translation parameter input 248 to specify the operation of the address translator 232. The translator 232 generates translated outputs A0', A1', and A2' 234 which are latched at the end of a decode stage in port address register 239. A number of different translation patterns are supported through use of a control input 244 which selects a particular translation operation. The values placed on the control input 244 may be based on a bit field in the instruction stored in the instruction register 231, such as an opcode or specific control bits as described at page 9, line 15 – page 10, line 15.

Claim 7

Claim 7 addresses a "processor register file indexing (RFI) address translation apparatus for translating an RFI sequence of instruction operand addresses to an RFI sequence of different operand addresses." As described at page 13, line 15 – page 14, line 4, an exemplary RFI VLIW processor 400 may be adapted as described in further detail in the discussions of Figs. 5 and 6 for use in conjunction with various embodiments of the present invention. Fig. 5, for example, shows further details of an ALU subsystem 500 of the RFI VLIW processor 400 of Fig. 4 with address translation functions in each operand address for translating RFI addresses, as described at page 15, line 22 – page 16, line 23. Fig. 6, for example, shows further details of address translation operation in RFI addressing in a data select unit (DSU) subsystem 600 of the RFI VLIW processor 400 of Fig. 4 as described at page 17, line 1 – page 19, line 3.

The apparatus of claim 7 comprises "a memory with an address input for selecting a data element from a plurality of data elements." The ALU subsystem 500 of Fig. 5 shows a memory as the register file 536 having address ports 544, 546, and 548 which are used to access operands from a block of operands stored in the register file 536 as described at page 16, lines 19-23. The DSU subsystem 600 of Fig. 6 described further details of the RFI address translation apparatus. As described at page 18, lines 15-17, register file 626 is a memory which receives operand addresses on Rt address port 624 for use in accessing operands from the register file 626.

The apparatus of claim 7 comprises "an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution." The DSU subsystem 600 of Fig. 6 shows an instruction register (IR) as DSU slot IR 602 receiving a DSU instruction from a VLIW DSU slot instruction bus which is one of the slot instruction buses from VLIW bus 434 of the RFI VLIW processor 400 of Fig. 4. As an example, the DSU slot IR 602 contains an instruction encoded with multiple fields such as Rt(5) field 614, a PEXCHG opcode, opcode extensions, and a Tsel field 642. The instruction is decoded in a pipeline decode stage 412 providing control information for the execution of the received instruction in a similar manner to the description of address translation in the processor 100 of Fig. 1 at page 7, lines 21-22, as indicated in the ALU subsystem 500 of Fig. 5 at page 16, lines 22-23, and as indicated in the DSU subsystem 600 of Fig. 6 at page 18, line 22 – page 19, line 3.

The apparatus of claim 7 comprises "an RFI update unit enabled to generate on the RFI update unit's output a linear sequence of RFI operand addresses in response to a received sequence of RFI translation type instructions." The DSU subsystem 600 of Fig. 6 shows an RFI update unit 610 which responds to an RFI enable signal and RFI parameter control information as described in further detail in Figs. 4, 5, and 6. Beginning with Fig. 4, the RFI

VLIW processor 400 describes a VLIW memory (VIM) controller 430 which determines an RFI enable signal during a predecode stage 408 for each of the RFI enabled XV instructions that access the same VIM 432 location as described at page 14, line 16 – page 15, line 2. Each time an RFI enabled XV instruction having the same VIM location is decoded, the RFI operation is enabled. The same VIM location indicates the same VLIW slot instructions, such as the ALU instruction received in the ALU slot IR 534 and the DSU instruction received in the DSU slot IR 602, are to be executed. For example, the ALU decode, RFI and translator control unit 530 responds to an RFI enable signal 532 generated in a VMC, such as VMC 430, in response to received execute VLIW (XV) instructions containing RFI control information as described at page 16, lines 7-19. Each execution of the same VLIW slot instruction causes the operand address to be updated via the RFI update unit 610 generating a linear sequence of operand addresses in response to the received sequence of RFI translation type instructions as described at page 17, lines 7-17 and at page 18, lines 17-19.

The apparatus of claim 7 also comprises "a multiplexer for selecting between the operand address from the instruction register for a first RFI operation and selecting the RFI update unit's output for subsequent RFI operations." Multiplexer 616 of DSU subsystem 600 of Fig. 6 is used to select on a first RFI operation the operand address 614 from the DSU slot IR 602 and on subsequent RFI operations to select updated operand addresses from a look ahead register 620 located in the RFI update unit 610 to generate a sequence of RFI operand addresses as described at page 17, line 17 – page 18, line 6.

The apparatus of claim 7 further comprises "an address translation unit for accessing the memory in a translation pattern, receiving a sequence of operand addresses from the multiplexer and, in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses in accordance with the translation pattern, the different operand addresses each accessing a data element from the memory through the address input." Rt address translator 634 of the DSU subsystem 600 of Fig. 6 receives the sequence of RFI operand addresses from the multiplexer 616 on output bus 630 and based on the translation type instruction translates the sequential stream of addresses to a desired address sequence on the translator bus 636. The translator bus 636 is latched in the port address register 618 at the end of each decode stage 412 for each RFI instruction executed and the different operand addresses are provided on the Rt address port 624 each accessing a data element of the register file 626. The translation pattern, as controlled by a specific set of {s,e} bits, is selected by use of the Tsel bit field 642. See, for example, the description at page 18, line 9 – page 19, line 3. The translation pattern may be determined from equations (1) and (2) based on the {s,e} bits as described at page 11, line 17 – page 12, line 11.

Claim 11

Claim 11 addresses an "address translation memory device for accessing data at translated addresses." As discussed at page 21, line 10 – page 22, line 3, a storage subsystem 800 shown in Fig. 8A includes a storage unit 810 which receives address inputs 815 and outputs data on read port output Rx 825 accessed at a translation of the address inputs 815.

The address translation memory device of claim 11 comprises a "first read address input." The address inputs 815 to the "storage unit 810, showing only a read port path" comprise the first read address input, as described at page 22, lines 4-5 and lines 8-10.

The address translation memory device of claim 11 also comprises a "storage device having data accessible at addressable locations, a second read address input internal to the address translation memory device for selecting data from the storage device during read operations, and a data output port." The storage subsystem 800 of Fig. 8A shows a storage device 835 which includes an address to location selector, such as 1 of 8 selector 840 having an address input A2' A1' A0', a storage array 845 with accessible locations R0-R7, and output Rx 825 as described at page 22, lines 17-19.

The address translation memory device of claim 11 further comprises "an address translation unit for accessing the storage device in a translation pattern, the address translation unit translating the first read address input in accordance with the translation pattern, to the storage device second read address input for reading data from the storage device at a translated address during a read operation." The storage subsystem 800 of Fig. 8A shows an address translator 830 which applies {s,e} information to translate the address input A2 A1 A0 to the address input A2' A1' A0' prior to accessing the storage device 835. The translation pattern is controlled by use of equation (2) for example using the {s,e} information as described at page 11, line 17 – page 12, line 11 and at page 22, lines 8-17.

Claim 16

Claim 16 addresses a "processor address translation method for translating an instruction operand address to a different operand address." As discussed at page 5, lines 16-18, a processor 100 shown in Fig. 1 may be adapted for use in conjunction with various embodiments of the present invention. Fig. 2D shows a processor subsystem 230 having an Rx read port translator 232 for translating instruction operand addresses to different addresses for reading addressable data from a register file 238 as described in detail at page 9, line 15 – page 10, line 15.

The method of claim 16 comprises "receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution." Instruction register (IR) 108 of processor 100 of Fig. 1 receives instructions from an instruction bus 106 that is coupled to a short instruction word memory 104. As an example, an arithmetic instruction is received in the IR 108 having three register file address fields, Rt 142, Rx 144, and Ry 146. The arithmetic instruction received in the IR 108 further supplies opcode and control bits 120 over an opA bus 154 to a decode and control unit 118 as described at page 6, lines 3-11 and at page 7, lines 11-22.

The method of claim 16 also comprises "translating the operand address according to a function." A block load with address translation instruction 300 of Fig. 3 shows an encoding of an instruction with a two bit translation selection (Tsel) field 305 that is used to select a translation pattern. A number of different translation patterns are supported through use of a control input 244 in Fig. 2D which selects a particular translation operation. The values placed on the control input 244 are based on a bit field in the instruction stored in the instruction register 231, such as the Tsel field 305 bits, as described at page 11, lines 3-16. The translation pattern, as controlled by a specific set of {s,e} bits, may be determined from the functions represented by equations (1) and (2) based on the {s,e} bits as described at page 11, line 17 – page 12, line 11 and at page 13, lines 12-14.

The method of claim 16 also comprises "accessing a data element with the translated address, and repeating the receiving, translating, and accessing steps to access data elements in a pattern according to the function." The block load with address translation instruction 300 of Fig. 3 may specify use of the Tsel field 305 to load a block of data from a linear sequence of data located in a local data memory to locations in a register file using bit-reverse operand addresses as described at page 11, lines 3-17. For each data element in the block of data, the steps of receiving, translating, and accessing are repeated until the block size, as specified by the block size field 315, is reached.

Claim 18

Claim 18 addresses an "address translation method for translating a first address of a first data element in a memory to a second address of a second data element in the memory." See, for example, Figs. 8A, 9, 10A, and 10B, page 21, line 10 – page 22, line 1, and page 24, line 10 – page 26, line 5 which describe translation operations including a transpose operation.

The method of claim 18 comprises "determining a set of {s, e} bits that specify a translation pattern." An {s,e} matrix 6 is determined for a transpose operation as described at page 25, line 14 – page 26, line 5. Other translation patterns may be specified by determining a different {s,e} matrix.

The method of claim 18 comprises "loading the set of {s, e} bits into an address translation parameter control register." The two port memory unit 900 of Fig. 9 shows a load translation pattern input 935 that is used to load the {s,e} bit translation state information as described at page 24, lines 13-16 and page 25, lines 16-17.

The method of claim 18 comprises "enabling an address translation unit for translation." Rx address translator 830 of storage subsystem 800 is enabled by the loading of translation parameters 820 which govern how the addresses access data from internal storage as described at page 21, line 22 – page 22, line 1. In a similar manner, the {s,e} matrix 6 is loaded into a parameter control register that is part of translator 940 in Fig. 9 to govern how data is to be accessed for the transpose operation as described at page 25, lines 16-18.

The method of claim 18 comprises "initiating a read operation to read a first data element at a first address during a read operation." As described at page 25, lines 18-20 in reference to Figs. 9 and 10B, when data is read the address may be supplied in a linear order as indicated in the left column as binary five bit addresses in table 1028.

The method of claim 18 also comprises "translating the first address to the second address in accordance with the {s, e} bit specified translation pattern." Equation (4) at page 19, lines 20-23 as specified with the {s,e} matrix 6 values is applied to the read address received at the read port address 925. This application of equation 4 translates the received address according to the specified transpose pattern of the {s,e} matrix 6 to a translated address.

The method of claim 18 further comprises "completing the read operation by reading the second data element at the second address." As shown in Fig. 10B and described at page 25, lines 14-20, an input read address, as specified from the left column of table 1028, is translated and data is then read as specified in the right column of table 1028 using matrix notation which is further illustrated in matrix 5 1024.

6. Grounds of Rejection to be Reviewed on Appeal

Claim 1 was rejected under 35 U.S.C. § 102(b) based on Saulsbury. Claim 2 was rejected under 35 U.S.C. § 103(a) as unpatentable over Saulsbury as applied to claim 1. Claims 3-19 were rejected under 35 U.S.C. § 103(a) as unpatentable over Saulsbury in view of Nair.

7. Argument

The final rejections under 35 U.S.C. §§ 102 and 103 did not follow M.P.E.P. §§ 706.02 and 706.02(j) which states respectively:

> For anticipation under 35 U.S.C. 102, the reference must teach every aspect of the claimed invention either explicitly or implicitly. Any feature not directly taught must be inherently present.

> After indicating that the rejection is under 35 U.S.C. 103, the Examiner should set forth...the difference or differences in the claim over the applied reference(s),...the proposed modification of the applied reference(s) necessary to arrive at the claimed subject matter, and ... an explanation why one of ordinary skill in the art at the time the invention was made would have been motivated to make the proposed modification.

As will be illustrated below, the claims of the present invention are not obvious in view of the references relied upon by the Examiner.

A. Section 102(b) Rejection of Claim 1

Claim 1 was rejected under 35 U.S.C. §102(b) as being anticipated by Saulsbury. Claim 1 recites "translating an instruction operand address to a different operand address." The instruction operand address is translated before accessing data as claimed in claim 1 "the different operand address accessing a data element from the memory through the address input".

Unlike the claimed invention, Saulsbury, when executing two transpose sub-instructions, first reads four source operand data values as specified by the operand addresses located in the transpose sub-instructions. These data values are then input to multiplexers which multiplexes the data values to select bit-fields from the four already read source registers, such as 16-bit fields in the illustrated example of Figs. 5, 6A, and 6B. The multiplexers generate two reordered sets of packed data values that are then stored in registers at destination addresses specified in the transpose sub-instructions. This approach is specifically shown in Saulsbury Fig. 5 in conjunction with Fig. 6A. For example, in Saulsbury Fig. 5, one of the four source register reg W 508-1 contains four data values a, b, c, and d, and another source register reg X 508-2 contains four data values e, f, g, and h. Four destination values are also shown in Fig. 5, with, for example, reg P 504-1 containing destination values a, e, i, and m. Four additional destination values are also shown in Fig. 5, with, for example, reg

R 504-3 containing destination values c, g, k, and o. The four registers 508-1, 508-2, 508-3, and 508-4 are shown in matrix 500 before performing the transpose operation and the four destination registers 504 are shown in transposed matrix 502 after executing multiple trans0 sub-instructions, such as the two trans0 sub-instructions 600 and 604 of Fig. 6A.

In Saulsbury's Fig. 6A Step 1, two instructions, trans0 RW, RX, RP 600 and trans0 RY, RZ, RR 604, are shown. The trans0 RW, RX, RP sub-instruction 600 specifies the source operand addresses RW and RX which are not translated and are used to directly read the source registers Rs1: W and Rs2: X. The source register fields of the trans register sub-instruction are shown in Fig. 4, which shows a first source register as Rs1 412, a second source register as Rs2 416, and a destination register Rd 420. Saulsbury, paragraph 45 states that the "first and second source addresses 412 and 416 are used to load a first and second source register" and the "destination address 420 is used to indicate where to store the results." Paragraph 45 further indicates that "all loads and stores are performed with the on-chip register file 60."

In the trans0 RW, RX, RP sub-instruction 600, the first source address (Rs1) W, as shown in Fig. 6A, specifies register W in the register file 60 in which register W contains data values a, b, c, and d that are the same data values as the contents of the register reg W 508-1 that exist prior to the transpose operation as shown in Saulsbury Fig. 5 matrix 500. In a similar manner, the second source address (Rs2) X, as shown in Fig. 6A, specifies register X in the register file 60 in which register X contains the values of e, f, g, and h that are the same data values as the contents of the register reg X 508-2 that exist prior to the transpose operation as shown in Fig. 5 matrix 500. The second trans0 RY, RZ, RR sub-instruction 604 specifies a third source address (Rs1) Y in the register file, as shown in Fig. 6A, which contains data values i, j, k, and l that are the same data values as the contents of the register reg Y 508-3 that exist prior to the transpose operation as shown in Fig. 5 matrix 500. Saulsbury's fourth source address (Rs2) Z, as shown in Fig. 6A, specifies register Z in the register file in which register Z contains data values m, n, o, and p that are the same data values as the contents of the register reg Z 508-4 that exist prior to the transpose operation as shown in Fig. 5 matrix 500.

The two trans0 instructions 600 and 604, when executed, first read the specified source registers Rs1: W, Rs2: X, Rs1: Y, and Rs2: Z without translating or changing the operand addresses Rs1 and Rs2 of the two trans0 sub-instructions 600 and 604. After the source operands have been read, the execution of the trans0 sub-instructions 600 and 604 cause the selection of specific data elements, in this case specific 16-bit data elements a-p, from the accessed source registers' packed data values. The selected data elements are then rearranged to form new packed data values. The rearranged new packed data values are then loaded in registers at the destination register addresses specified in the two trans0 instructions. Saulsbury Figs. 5 and 6A and paragraphs 53-55.

The execution of two trans0 sub-instructions 600 and 604 is further shown in Fig. 7 and described in paragraphs 59 and 60. In paragraph 59, Saulsbury describes "the first and second source registers 508-1, 508-2 and the first destination register 504-1 are addressed by a first trans0 sub-instruction 600. Likewise, the third and fourth source registers 508-3, 508-4 and the third destination register 504-3 are addressed by a second trans0 sub-instruction 604." In paragraph 60, Saulsbury describes "inputs coupled to the source registers 508 and outputs coupled to the destination registers 504. The instruction processor 700 also includes multiplexers or the like, that implement the redirection of data from the source registers 508 to the appropriate destination registers".

Thus, the instruction operand source register addresses and the destination register addresses **are not translated** to different operand addresses in Saulsbury's approach. To the contrary, the source register address fields are used directly to read the specified registers at the address supplied by the trans0 sub-instructions. The destination addresses are used directly to store the results at the address supplied by the trans0 sub-instructions. Saulsbury does not teach and does not make obvious an apparatus or method as presently claimed which addresses the problem of translating an instruction operand address to a different operand address. Claim 1 is not taught, is not inherent, and is not obvious in light of Saulsbury.

B. Rejections under Section 103(a)

These rejections are not supported by the relied upon art. 35 U.S.C. § 103 which governs obviousness indicates that "differences between the subject matter sought to be patented and the prior art" are to be assessed based upon "the subject matter as a whole". Analyzing the entirety of each claim, the rejections under 35 U.S.C. § 103 are not supported by the relied upon art as addressed further below. Only after an analysis of the individual references has been made can it then be considered whether it is fair to combine teachings. However, as addressed further below, fairness requires an analysis of failure of others, the lack of recognition of the problem, and must avoid the improper hindsight reconstruction of the present invention. Such an analysis should consider whether the modifications are actually suggested by the references rather than assuming they are obvious. The 35 U.S.C. § 103 rejections made here pick and choose elements from two separate references, neither of which presents any motivation for making the suggested combination. This approach constitutes impermissible hindsight and must be avoided. As required by 35 U.S.C. § 103, claims must be considered as a whole. When so considered, the present claims are not obvious.

Claim 2

Dependent claim 2 was rejected under 35 U.S.C. § 103(a) as unpatentable over Saulsbury as applied to claim 1. As addressed above, Saulsbury does not teach and does not make obvious the processor address translation apparatus of claim 1. Since claim 2 depends from and contains all the limitations of claim 1, claim 2 distinguishes from Saulsbury in the same manner as claim 1.

Additionally, the Examiner cites paragraph 67 on page 6 of Saulsbury as purportedly describing "a plurality of translation parameters and address translation functions supporting a plurality of translation patterns" as claimed in claim 2. The table associated with paragraph 67, illustrates the type of VLIW processor, the width of the data elements, and the width of the registers that are compatible to obtain the "sixteen element matrix" associated with each variation in the table. "The sole table indicates some of the possible variations of this invention for performing a transpose in one issue... All the variations in the table presume a sixteen-element matrix." As an example, Saulsbury describes the 6th row of the table "For example, a two way VLIW architecture with two processing paths and sixteen bit wide elements in one hundred and twenty-eight bit wide registers could perform a sixteen-element transpose in one issue." Saulsbury, paragraph 67. The table does not describe "a plurality of translation parameters and address translation functions supporting a plurality of translation patterns" as presently claimed in claim 2 since the table presents only variations in processor and data organizations to support a single transpose operation. Claim 2 is not taught, is not inherent, and is not obvious in light of Saulsbury.

Claims 3-19

Claims 3-19 were rejected under 35 U.S.C. §103(a) based on Saulsbury as applied to claim 2 and in view of Nair.

Claims 3-6

Since claims 3-6 depend from and contain all the limitations of claim 1, these claims distinguish from the references in the same manner as claim 1 addressed above.

Additionally regarding claim 3, the Examiner cites column 12, table 1 of Nair as purportedly describing an apparatus for matrix processing that can manipulate address bits using matrix multiply, addition, subtraction, bit-reverse operations, etc. To the contrary, Nair, column 12, table 1, is a listing of specific matrix operations that can be performed on data elements of a matrix. The matrix data elements, as shown in Nair Fig. 2, are "stored as packed data sets in physical register set 220 and then mapped to logical register set 210. Once mapped to logical register set 210, the various elements may be manipulated as matrix elements." Nair, col. 6. lines 23-28. Nair then describes the operations of table 1 additionally including logical operations as being operations that are "performed on the various matrix elements." Nair, col. 12, lines 14-22. Nair does not describe operations that are performed on an "operand address input" to produce a "translated address output" according to specific equations with "k by k s bits and k e bits" as presently claimed in claim 3. Nair fails to cure the deficiencies of Saulsbury as addressed above.

Claim 7

Regarding claim 7, the Examiner incorporates the rationale of the rejection of claims 1 and 2. As noted above, the rationale for the rejection of claims 1 and 2 is faulty. For example, Saulsbury's DRAM bank #32-1 is not a memory addressed by "an address translation unit for accessing the memory in a translation pattern" as claimed in claim 7, but rather is just illustrated to provide Saulsbury's system context. Further, Saulsbury's Fig. 3 does not illustrate "an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution" as claimed in claim 7, but rather indicates data types as stated in the descriptive labels under each data type depicted in Fig. 3 and as described in Saulsbury's paragraph 40. Also, Saulsbury's paragraphs 45, 46, 51, and 52 do not describe a register file indexing address translation apparatus. Saulsbury's paragraph 45 describes a matrix transpose instruction format. Saulsbury's paragraph 46 describes a limitation that two transpose sub-instructions are issued at a time and the processing paths associated with both sub-instructions have access to the register files associated with the processing paths. Paragraph 51 describes a data organization matrix 500 as stored in a register file prior to performing the transpose operation and a transposed matrix 502 data organization as stored in a register file after performing the transpose operation. Paragraph 52 suggests that larger matrices may be broken into "four-by-four chunks and manipulated separately". None of the cited paragraphs, 45, 46, 51 or 52, describe, for example, a "register file indexing (RFI) address translation apparatus" where a "sequence of RFI operand addresses" are generated by an "RFI update unit" and the "sequence of RFI operand addresses" are translated "to form a sequence of different operand addresses in accordance with the translation pattern" as presently claimed in claim 7. Claim 7 is not taught, is not inherent, and is not obvious in light of Saulsbury.

Claims 8-10

Since claims 8-10 depend from and contain all the limitations of claim 7, these claims distinguish from the references in the same manner as claim 7 addressed above.

Claim 11

The Examiner rejects claim 11 based on the same rationale as of claim 1. As noted above, the rationale for the rejection of claim 1 is faulty. For example in Saulsbury's Fig. 2, an "instruction decode and issue logic stage 58 of the processing core pipeline 50 receives VLIW instruction word 52 and decodes and issues the sub-instructions 54 to the appropriate processing paths 56." Saulsbury, paragraph 35. The instruction decode and issue logic stage 58 of the processing core pipeline 50 decodes and issues the sub-instructions 54 which are operations that are clearly different than the "address translation unit translating the first read address input in accordance with the translation pattern, to the storage device second read address input for reading data from the storage device at a translated address during a read operation" as claimed in claim 11. The "address translation memory device" of claim 11 is not taught, is not inherent, and is not obvious in light of Saulsbury.

Claims 12-15

Since claims 12-15 depend from and contain all the limitations of claim 11, these claims distinguish from the references in the same manner as claim 11 addressed above.

Claim 16

Clarification was requested from the Examiner as to how independent claim 16 is rejected based on the same rationale as the rejection of the amended dependent claim 4. Claim 4 claims "the instruction is a block load instruction" and claim 16 claims a "processor address translation method". The rationale for rejecting claim 4 is not supported as addressed above. If the rejection of claim 16 is truly based on the rationale for rejecting claim 4 it should be withdrawn on the same basis. As claim 16 was rejected without any supporting rational for the rejection, it is somewhat difficult to directly rebut the reasoning of the Examiner. Nonetheless, it is clear that the relied upon references do not anticipate and do not render obvious the processor address translation method as claimed in claim 16. Claim 16 is not taught, is not inherent, and is not obvious in light of Saulsbury.

Such a rejection is not in accordance with M.P.E.P. § 706.02(j) which requires that the Examiner set out "an explanation why one of ordinary skill in the art at the time the invention was made would have been motivated to make the proposed modification." (emphasis added)

Claims 17 and 19

Since claims 17 and 19 depend from and contain all the limitations of claim 16, these claims distinguish from the references in the same manner as claim 16 addressed above.

Claim 18

Claim 18 was rejected based on the same rationale as the rejection of claims 1 and 3. As described above, that analysis is not supported by Saulsbury. As claim 18 was rejected without any further supporting rational for the rejection, it is somewhat difficult to directly rebut the reasoning of the Examiner. Nonetheless, it is clear that the relied upon references do not anticipate and do not render obvious the processor address translation method as claimed in claim 18. Claim 18 is not taught, is not inherent, and is not obvious in light of Saulsbury.

Overall, Applicant is somewhat puzzled by the Examiner's response to the previously submitted arguments and the apparent refusal of the Examiner to consider both the plain language and the context of the present claims. The relied upon references do not teach and do not render obvious claims 1-19.

To sum up, Saulsbury and Nair do not show and do not suggest address translation apparatuses and methods for translating an address to a different address according to a translation pattern or a function as presently claimed. Nothing in the cited references indicates a recognition of the problems addressed by the present invention. Further, nothing in the cited references indicates apparatuses or methods which would solve the problems addressed by the present invention. The claims of the present invention are not taught, are not inherent, and are not obvious in light of the art relied upon.

C. The Examiner's Findings of Obviousness are Also Contrary to Law of the Federal Circuit

As shown above, the invention claimed is not suggested by the relied upon prior art. The references cited by the Examiner, if anything, teach away from the present invention. It is only in hindsight, after seeing the claimed invention, that the Examiner could combine the references as the Examiner has done. This approach is improper under the law of the Federal Circuit, which has stated that "[w]hen prior art references require selective combination by the Court to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gleaned from the invention itself." Uniroyal, Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 1051, 5 U.S.P.Q. 2d 1434, 1438 (Fed. Cir. 1988), cert. den., 109 S. Ct. 75, 102 L.Ed. 2d 51 (1988); quoting Interconnect Planning Corp. v. Feil, 774 F.2d 1132, 1132, 227 U.S.P.Q. 543, 535 (Fed. Cir. 1985). Furthermore, "[i]t is impermissible to use the claims as a frame and the prior art references as a mosaic to piece together a facsimile of the claimed invention." Uniroyal, 837 F.2d at 1051, 5 U.S.P.Q. 2d at 1438. Similarly, "[t]he mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification." In re Laskowski, 871 F.2d 115, 117, 10 U.S.P.Q. 2d 1397, 1398 (Fed. Cir. 1989), quoting In re Gordon, 733 F.2d 900, 902, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984). No such suggestion is found here.

In addition, the Examiner does not appear to have considered "where the references diverge and teach away from the claimed invention", Akzo N.V. v. International Trade

Commission, 808 F.2d 1471,.1481, 1 U.S.P.Q. 2d 1241, 1246 (Fed. Cir. 1986), cert. den., 107 S. Ct. 2490, 482 U.S. 909, 107 S.Ct. 2490 (1987); and W.L. Gore Associates, Inc., 721 F.2d 1540, 220 U.S.P.Q. 303 (Fed. Cir. 1983); nor has the Examiner read the claims as a whole, as required by statute. 35 U.S.C. §103. See also, Smithkline Diagnostics Inc. v. Helena Laboratories Corp., 859 F.2d 878, 885, 8 U.S.P.Q. 2d 1468, 1475 (Fed. Cir. 1988); and Interconnect Planning Corp., 774 F.2d at 1143, 227 U.S.P.Q. at 551.

In In re Laskowski, 871 F.2d 115, 10 U.S.P.Q. 2d 1397, the Federal Circuit reversed an obviousness rejection of the claims in an application for a bandsaw. The claimed bandsaw used a pulley type wheel loosely fitted with a tire. The primary reference showed a similar bandsaw where the band was tightly fitted. The Federal Circuit stated that the prior art did not provide a suggestion, reason or motivation to make the modification of the reference proposed by the Commissioner. Id. at 1398. The Court added that "there must be some logical reason apparent from the positive, concrete evidence of record which justifies a combination of primary and secondary references." Id. quoting In re Regel, 526 F.2d 1399, 1403, 188 U.S.P.Q. 136, 139 (C.C.P.A. 1975), citing In re Stemniski, 444 F.2d 581, 170 U.S.P.Q. 343 (C.C.P.A. 1971).

In Uniroyal Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 5 U.S.P.Q. 2d 1434 (Fed. Cir. 1988), cert. den., 109 S. Ct. 75, 102 L.Ed. 2d 51 (1988), the Federal Circuit reversed the District Court's finding that the claims for a patent for an air flow deflecting shield were obvious. Without any suggestion in the art, the District Court improperly chose features from several prior art references to recreate the claimed invention.

The Examiner's rejection suggests that the Examiner did not consider and appreciate the claims as a whole. The claims disclose a unique combination with many features and advantages not shown in the art. It appears that the Examiner has oversimplified the claims and then searched the prior art for the constituent parts. Even with the claims as a guide, however, the Examiner did not recreate the claimed invention.

8. Conclusion

The rejection of claims 1-19 should be reversed and the application promptly allowed.

Respectfully submitted,

Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713
(919) 806-1600

CLAIMS APPENDIX
(Claims Under Appeal)

1. A processor address translation apparatus for translating an instruction operand address to a different operand address, the processor address translation apparatus comprising:

a memory with an address input for selecting a data element from a plurality of data elements;

an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution; and

an address translation unit for accessing the memory in a translation pattern, having the operand address as input and, in response to the instruction received in the instruction register, translating the operand address to form the different operand address in accordance with the translation pattern, the different operand address accessing a data element from the memory through the address input.

2. The processor address translation apparatus of claim 1 wherein the address translation unit further comprises:

a plurality of translation parameters and address translation functions supporting a plurality of translation patterns; and

an input to select a translation pattern from the plurality of supported translation patterns.

3. The processor address translation apparatus of claim 2 wherein the translation parameters include k by k s bits and k e bits for a k bit address and address translation functions further comprises combinatorial logic governed by the following equations, where the operand address input is A0, A1, ..., A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and translated address output are A0', A1', ..., A(k-1)',

$$\begin{pmatrix} A0' \\ A1' \\ \vdots \\ A(k-1)' \end{pmatrix} = \begin{pmatrix} s0 & s1 & \cdots & s(k-1) & e0 \\ sk & s(k+1) & \cdots & s(2k-1) & e1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(k-1)k & s(k-1)k+1 & \cdots & s(k-1)(k+1) & e(k-1) \end{pmatrix} \times \begin{pmatrix} A0 \\ A1 \\ \vdots \\ A(k-1) \\ 1 \end{pmatrix}.$$

4. The processor address translation apparatus of claim 1 wherein the instruction is a block load instruction.

5. The processor address translation apparatus of claim 1 disposed within a plurality of instruction operand address paths for a plurality of instructions, the plurality of instructions fetched for simultaneous execution.

6. The processor address translation apparatus of claim 5 wherein the plurality of instructions constitute a very long instruction word (VLIW).

7. A processor register file indexing (RFI) address translation apparatus for translating an RFI sequence of instruction operand addresses to an RFI sequence of different operand addresses, the processor RFI address translation apparatus comprising:

a memory with an address input for selecting a data element from a plurality of data elements;

an instruction register for receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution;

an RFI update unit enabled to generate on the RFI update unit's output a linear sequence of RFI operand addresses in response to a received sequence of RFI translation type instructions;

a multiplexer for selecting between the operand address from the instruction register for a first RFI operation and selecting the RFI update unit's output for subsequent RFI operations; and

an address translation unit for accessing the memory in a translation pattern, receiving a sequence of operand addresses from the multiplexer and, in response to the sequence of RFI operand addresses, translating the sequence of RFI operand addresses to form a sequence of different operand addresses in accordance with the translation pattern, the different operand addresses each accessing a data element from the memory through the address input.

8. The processor RFI address translation apparatus of claim 7 disposed within PEs of an array of PEs.

9. The processor RFI address translation apparatus of claim 7 disposed within a plurality of instruction operand address paths for a plurality of instructions, the plurality of instructions fetched for simultaneous execution.

10. The processor RFI address translation apparatus of claim 9 wherein the plurality of instructions constitute a very long instruction word (VLIW).

11. An address translation memory device for accessing data at translated addresses, the address translation memory device comprising:

a first read address input;

a storage device having data accessible at addressable locations, a second read address input internal to the address translation memory device for selecting data from the storage device during read operations, and a data output port; and

an address translation unit for accessing the storage device in a translation pattern, the address translation unit translating the first read address input in accordance with the translation pattern, to the storage device second read address input for reading data from the storage device at a translated address during a read operation.

12. The address translation memory device of claim 11 further comprises:

a first write address input;

a storage device having data accessible at addressable locations, a second write address input for selecting data in the storage device during write operations, and a data input port; and

an address translation unit, for accessing the storage device in a translation pattern, the address translation unit translating the first write address input in accordance with the translation pattern, to the storage device second write address input for writing data to the storage device at a translated address during a write operation.

13. The address translation memory device of claim 11 wherein the storage device further comprises location selection logic merged with the address translation unit.

14. The address translation memory device of claim 11 further comprises:

a plurality of translation parameters and address translation functions supporting a plurality of translation patterns; and an input to select a translation pattern from the plurality of supported translation patterns.

15. The address translation memory device of claim 14 wherein the translation parameters include k by k s bits and k e bits for a k bit address and address translation functions further comprises combinatorial logic governed by the following equations, where the first read address input is A0, A1, ..., A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and translated address output are A0', A1', ..., A(k-1)',

$$\begin{pmatrix} A0' \\ A1' \\ \vdots \\ A(k-1)' \end{pmatrix} = \begin{pmatrix} s0 & s1 & \cdots & s(k-1) & e0 \\ sk & s(k+1) & \cdots & s(2k-1) & e1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(k-1)k & s(k-1)k+1 & \cdots & s(k-1)(k+1) & e(k-1) \end{pmatrix} \times \begin{pmatrix} A0 \\ A1 \\ \vdots \\ A(k-1) \\ 1 \end{pmatrix}.$$

16. A processor address translation method for translating an instruction operand address to a different operand address, the address translation method comprising:

receiving an instruction encoded with an operand address and control information indicating the operand address is to be translated as part of the instruction's execution;

translating the operand address according to a function; and

accessing a data element with the translated address, and repeating the receiving, translating, and accessing steps to access data elements in a pattern according to the function.

17. The processor address translation method of claim 16 wherein the function comprises combinatorial logic for translating the operand address.

18. An address translation method for translating a first address of a first data element in a memory to a second address of a second data element in the memory, the address

translation method comprising:

determining a set of {s, e} bits that specify a translation pattern;

loading the set of {s, e} bits into an address translation parameter control register;

enabling an address translation unit for translation;

initiating a read operation to read a first data element at a first address during a read operation;

translating the first address to the second address in accordance with the {s, e} bit specified translation pattern; and

completing the read operation by reading the second data element at the second address.

19. The address translation method of claim 16 wherein the set of {s, e} bits include k by k s bits and k e bits for a k bit address and address translation functions further comprises combinatorial logic governed by the following equations, where the operand address is A0, A1, ..., A(k-1), product operations are treated as ANDs, sum operations are treated as XORs, and translated address output are A0', A1', ..., A(k-1)',

$$\begin{pmatrix} A0' \\ A1' \\ \vdots \\ A(k-1)' \end{pmatrix} = \begin{pmatrix} s0 & s1 & \cdots & s(k-1) & e0 \\ sk & s(k+1) & \cdots & s(2k-1) & e1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(k-1)k & s(k-1)k+1 & \cdots & s(k-1)(k+1) & e(k-1) \end{pmatrix} \times \begin{pmatrix} A0 \\ A1 \\ \vdots \\ A(k-1) \\ 1 \end{pmatrix}.$$

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.